TASK FORCE WEB

Navy Enterprise Portal Architecture

9 July 2002

Michael Donovan, EDS
NMCI/ISF Enterprise Technology

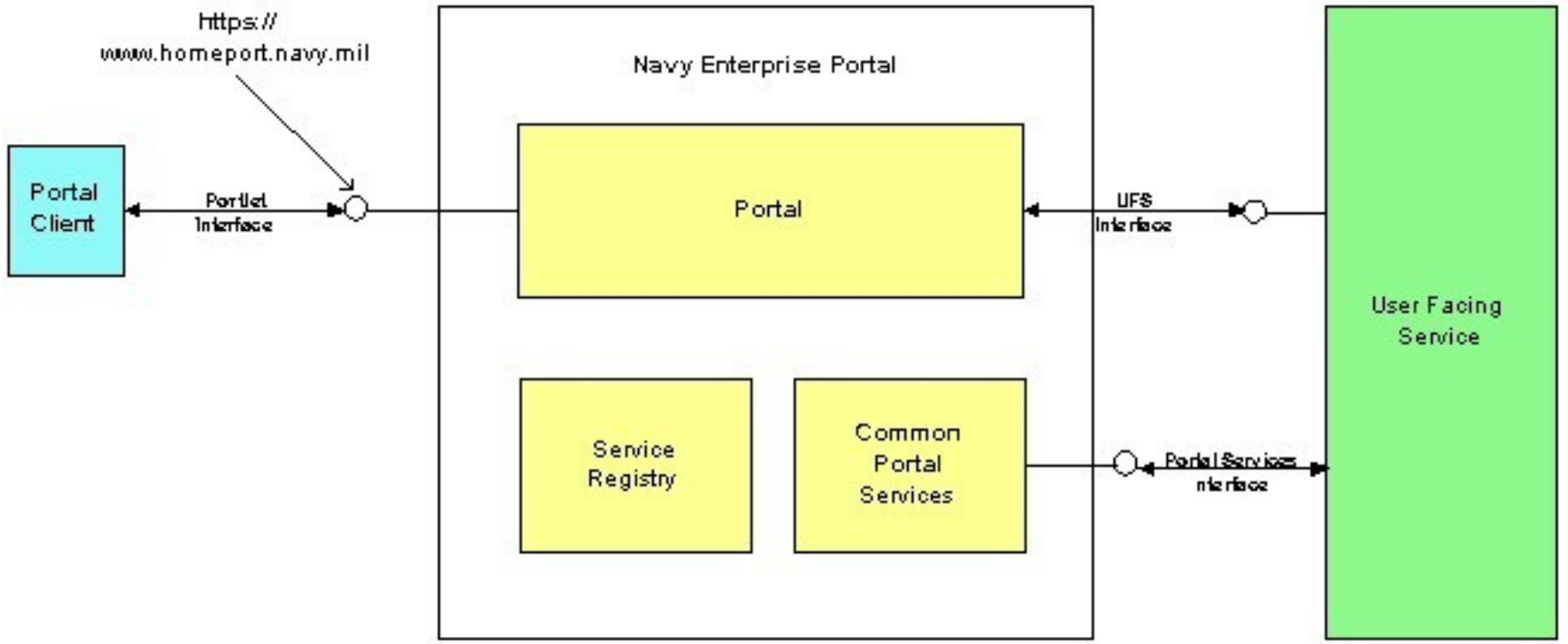
Agenda

- **Services**
- **Navy Enterprise Portal Architecture**
- **Portal Components**
- **Execution Sequence**
- **Standards**
- **What's Changed?**

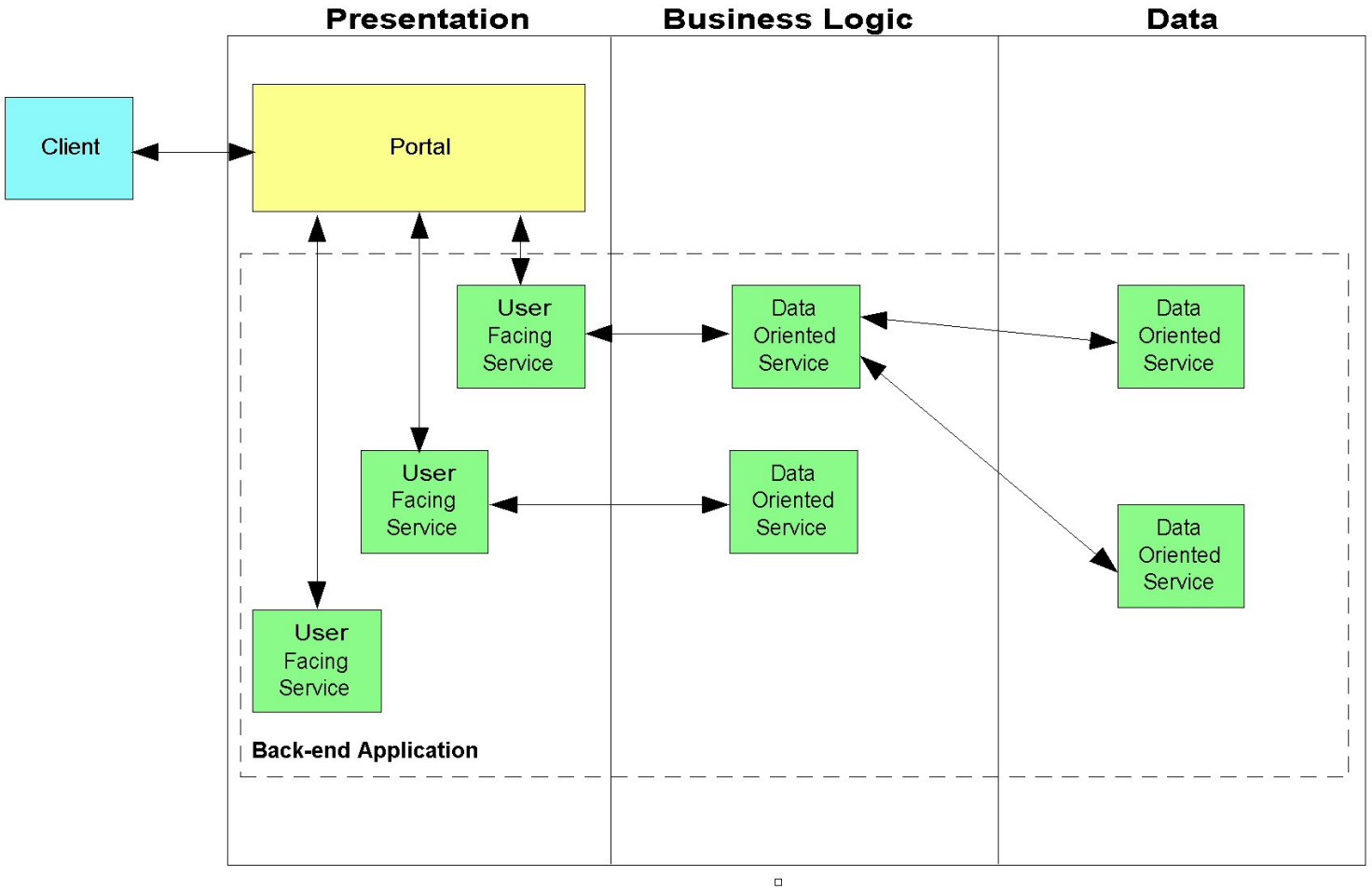
What is a Service?

- A *Service* is a software component that is registered and provides access to information or functionality
  - *User Facing Service (UFS)* implements presentation layer of application, such as portlet or data display
  - *Data Oriented Service (DOS)* implements business logic or data access service, normally outputs XML

NEP Architecture

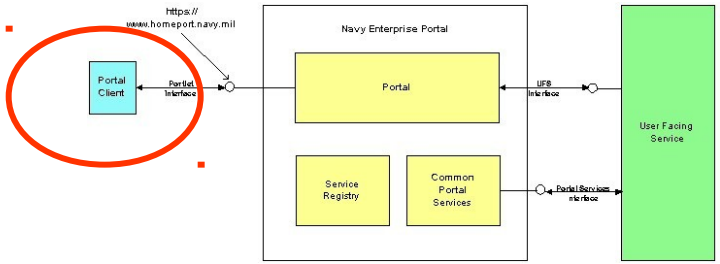


n-Tier Architectures



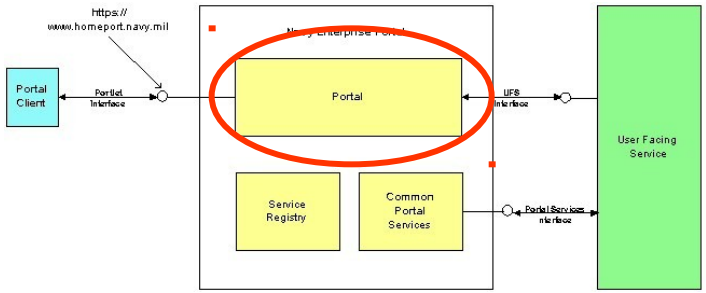
Portal Components

- A *Client* is a device capable of interaction with the portal
  - Commonly web browser, standard is Internet Explorer 5.5. Netscape and others are supported, but with limitations
  - Strategy is to apply XSL to XML data based on device type to extend functionality to other devices



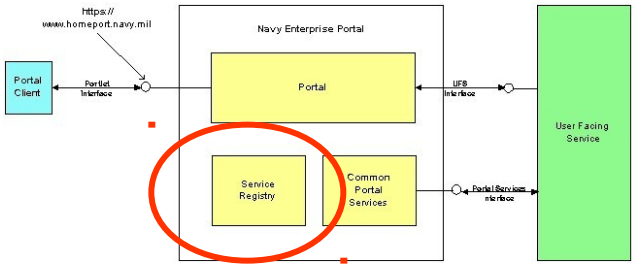
Portal Components

- The *Portal Engine* aggregates content from distributed user facing services
  - Provides personalization and customization
  - Provides common identity authentication
  - Provides common look and feel
  - Strategy is to support common industry standards as they develop, current implementation is GOTS



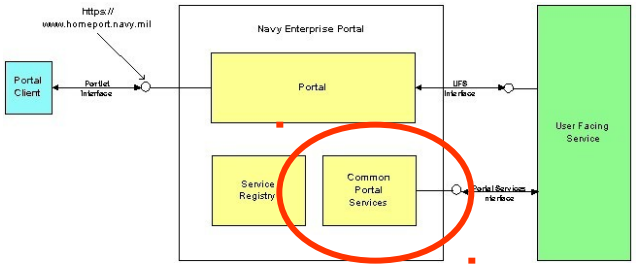
Portal Components

- The *Service Registry* is a private, globally distributed registry of web services information.
  - All services (UFS and DOS) must be registered
  - Strategy is to implement private UDDI V2.0 compliant registry using a COTS product



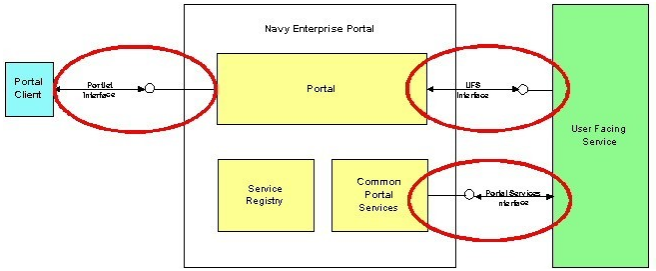
Portal Components

- *Portal Services* are exposed to enable callbacks from user facing services
  - Services are:
    - Service Registry (when UDDI implementation complete)
    - URL Rewrite
    - Web Service Execution
    - XML Transformation
  - Strategy is to migrate interfaces to industry standards as they evolve
    - WSRP, UDDI

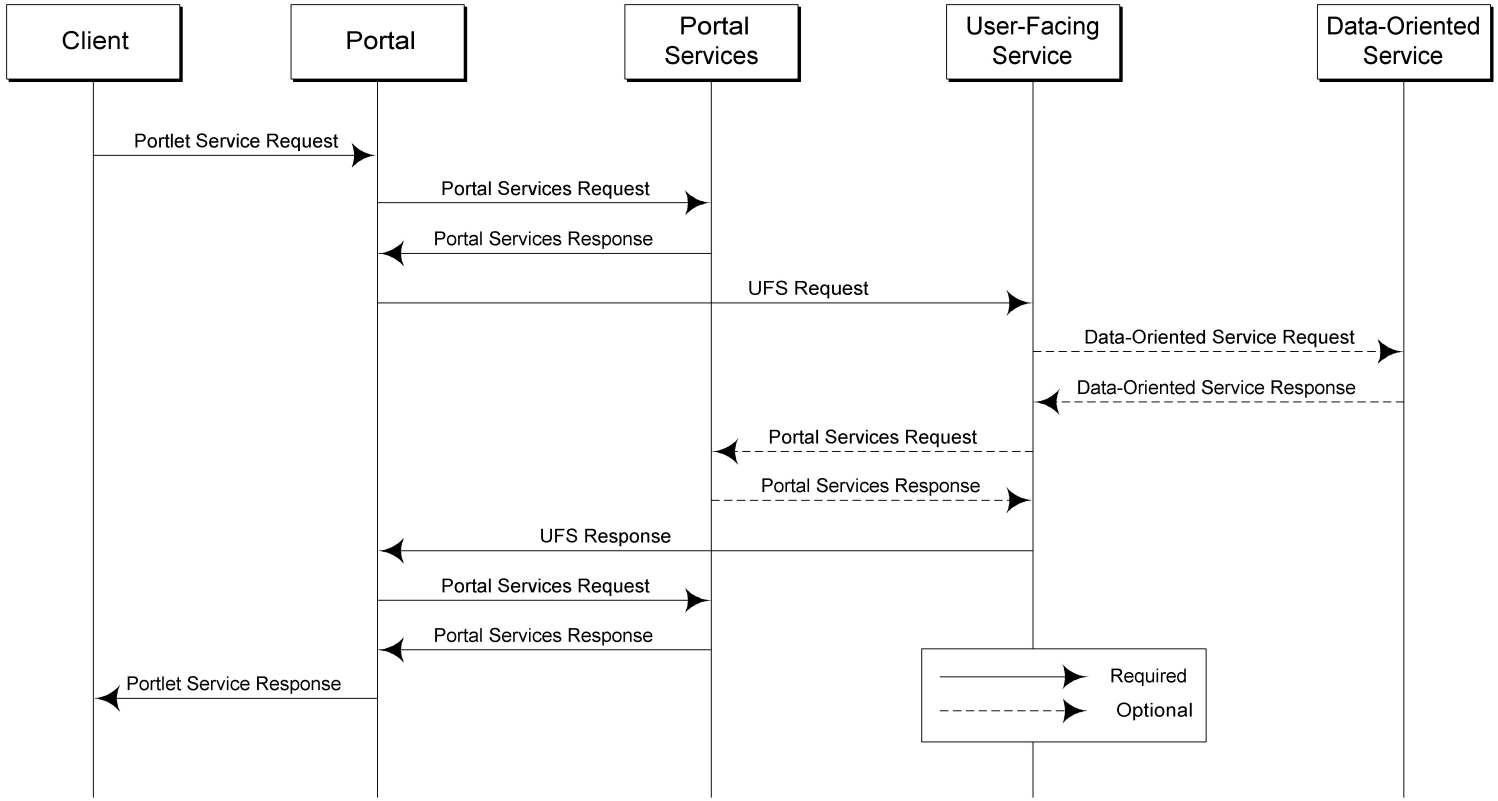


Portal Interfaces

- An *Interface* is a common boundary between independent systems or modules where interactions take place
  - The portal interfaces represent a set of rules, guidelines, standards, and protocols that must be followed to enable successful interaction across the common boundary.
  - The portal interfaces that developers must understand are:
    - *Portlet Interface*
    - *UFS Interface*
    - *Portal Services Interface*



Execution Sequence



Standards

- The NEP is based on COTS technology and industry standards
- GOTS technology is minimized and when used, is because either:
  - No clear standard exists, so a transition strategy has been implemented (Portal Connector)
  - Protection of Navy intellectual capital requires a specific GOTS component to implement a unique process

Current Web Standards

- Currently published by recognized standards body, e.g., OASIS, W3C, and ISO
- Expected to be used for portal content integration, when appropriate

Standard	Body / Latest Specification
HyperText Transfer Protocol (HTTP) 1.1	IETF, June 1999
HyperText Markup Language (HTML) 4.01	W3C, December 1999
ECMAScript (Javascript)	ECMA, December 1999
Cascading Style Sheets, level 1 (CSS1)	W3C, January 1999
Extensible Markup Language (XML) 1.0	W3C, October 2000
Namespaces in XML	W3C, January 1999
Extensible Stylesheet Language (XSL) 1.0	W3C, October 2001
XSL Transformations (XSLT) 1.0	W3C, November 1999
Simple Object Access Protocol (SOAP) 1.1	W3C, May 2000
Web Services Description Language (WSDL) 1.0	W3C, March 2001

Emerging / Future Standards

- Emerging or currently in development by standards bodies
- Probably will be used for portal integration in the future

Standard	Body / Latest Specification
Web Services for Remote Portals (WSRP)	OASIS, n/a
Extensible HyperText Markup Language (XHTML) 1.0	W3C, June 2002
Scalable Vector Graphics (SVG) 1.1	W3C, April 2002
Cascading Style Sheets, level 2 (CSS2)	W3C, May 1998
Simple Object Access Protocol (SOAP) 1.2	W3C, June 2002

What's Changed?

Lesson Learned	Action Taken
Integration complexity	•Developer guidance modified to encapsulate portal internal functions. •EMS Servers eliminated and functions distributed •Focus on service interaction, not implementation •Exposed more portal functions through API's to enhance flexibility
Service Modules purpose not clear	•Redefined as portlet implemented through UFS •Difference between UFS and DOS described
Single Sign-On (SSO) complexity	•Common Identity defined •Intra-domain SSO implementation •Clearer development path described •Inter-domain strategy defined